Driving on the Surface of Mars with the Rover Sequencing and Visualization Program

J. Wright, F. Hartman, B. Cooper, S. Maxwell, J. Yen, J. Morrison

Jet Propulsion Laboratory, 4800 Oak Grove Dr., Pasadena, CA, USA

John.R.Wright@jpl.nasa.gov

Abstract: Operating a rover on Mars is not possible using teleoperations due to the distance involved and the bandwidth limitations. Instead, rovers are autonomous and go about their operations in an unsupervised mode. To operate these rovers requires sophisticated tools to make operators knowledgeable of the terrain, hazards, features of interest, and rover state and limitations, and to support building command sequences and rehearsing expected operations. The primary requirement is to rapidly convey the optimal level of situational awareness to the operators to support daily cycles of operations and commanding. This paper discusses how the Rover Sequencing and Visualization Program and a small set of associated tools support this requirement.

Keywords: Mars, rovers, autonomy, commanding, ground operations, telepresence.

1. Introduction

The Jet Propulsion Laboratory has been operating twin rovers on Mars since January of 2004. The two rovers, named Spirit and Opportunity and part of the Mars Exploration Rovers (MER) mission, are exploring Gusev Crater [LEGER05] and Meridiani Planum [BIESIADECKI05] respectively. Because the rovers are solar-powered, their operational cycles are tied to the Martian diurnal cycle. Consequently, Earth-bound operators are also tied to the Martian day, which is about 40 minutes longer than an Earth day. The similarity in the length of the day on the two planets, as well as the desire to maximize science return while also maintaining the safety of the rovers, leads to several operational constraints. The first is that to maximize science return for the mission, activities must occur every sol, or Martian day. However, after the rover traverses to a new area, the final position is not precisely known, requiring data downlink to analyze the current rover state for safely planning future operations. Thus, an additional constraint is that the operators must wait for data to plan future activities. The distance between Earth and Mars, which changes throughout the year, provides additional constraints by precluding teleoperation of the rovers due to bandwidth and lag time limitations. Thus, the rovers are autonomous and proceed with their scheduled activities without intervention or supervision from the ground. These constraints combine to essentially mandate a daily mission planning cycle in which downlinked data is processed, the rover state is analyzed, the terrain is studied for hazards, traversability, and features of interest, and new commands are uplinked every sol.

The Rover Sequencing and Visualization Program (RSVP) is a set of software tools that are designed to work in this daily planning environment to support the rapid analysis of rover state information and creation of command sequences. Previous planetary missions could plan for encounters known months or years in advance. The Mars Pathfinder mission pioneered the use of the daily planning cycle [COOPER98] and this paradigm was extended for MER. The RSVP tools include terrain model visualization and interaction, numeric data plotting and analysis, image display and interrogation, command sequence visualization, sequence rehearsal, kinematic modelling of rover and terrain interactions, and time-based modelling of spacecraft and planetary bodies for analysis of communication issues, incident solar energy, and shadowing.

Essentially, RSVP must provide the operators with information on the local terrain and rover state in a mode that supports rapid assimilation and understanding. Then it must support rapid assembling and testing of command sequences for correct execution. It must also support the creation of documentation and archival products.

2. Data Analysis

RSVP ingests several types of data to support its mission. The types of data include current and historical state information, imagery with associated state information, three-dimensional terrain models, and science activity plans. Figure 1 contains a diagram illustrating the flow of data into RSVP, with some overview of the sources and processing of that data in the ground data system.

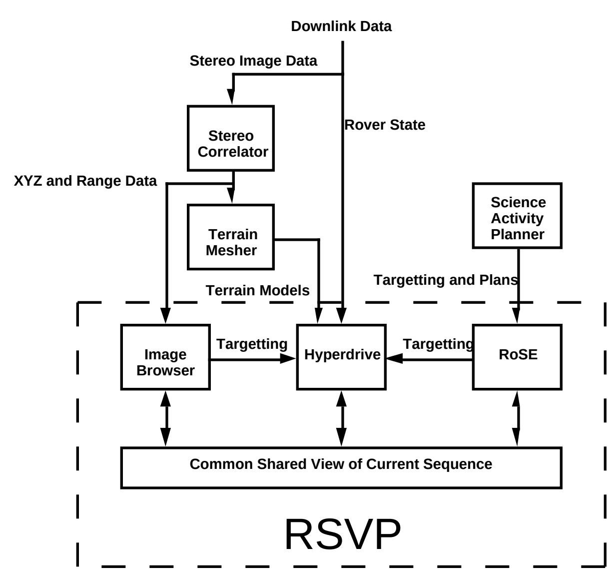


Figure 1 – Data Sources for RSVP

The state information is used to analyze and review the current state of the rover, identify any anomalous issues, review previously commanded activities, and verify that the rover is ready to accept and perform a new set of commanded activities. Imagery is used to visually examine the surrounding terrain, identify hazards and areas of interest, specify targeting information for additional observations, and essentially present the operator with the least processed view of the current environment. The terrain models are used similarly but with full three-dimensional representations of the terrain and unlimited viewing from any position and orientation. The science activity plans are produced by the science team and are used as guides for planning approaches to targets of interest and for defining desired activities for the instruments on the Instrument Deployment Device (IDD), the robotic arm mounted to the front of each rover.

2.1 State Analysis

The state of the rover is represented by numeric values collected by numerous sensors located at various points on the rover, as well as the results of internal calculations based on the sensor data. Examples of data channels (the data associated with a single sensor) include the rocker and bogey joint angles of the suspension, the wheel steering angles, the wheel rotation counters, the joint angle sensors on the arm, temperature sensors, and many others. Examples of internally computed channels include the XYZ location of the rover. As the rover traverses the terrain, its inertial system maintains good tracking of orientation. By using orientation and wheel rotations, the rover can estimate its position fairly well when traversing relatively flat terrain. However, when the rover gets into a region of high slip, this "dead reckoning" approach breaks down. A different method of determining position, called visodom or visual odometry, is used here, which must be commanded, and will be discussed in a later paragraph. Another example of internally calculated state information is error conditions detected. Examples of these include joint or current limits exceeded, potential IDD and rover body collision detection, inability to reach a commanded destination, and others.

Figure 2 – Numeric Display of Rover State Information

Analysis of the state information takes several forms. The first is a simple display of all the numeric values, as seen in Figure 2. Note that there are actually several pages of displays such as this, each focussing on a particular subsystem such as mobility or IDD. There are also summary pages that present the most pertinent information for multiple subsystems. This type of display has only limited usefulness because it represents only a single snapshot in time (typically the latest or current rover state) and is so full of detailed information that it is difficult to locate pertinent data without a great deal of experience. Some functions do aid the operator by color-coding particular pieces of information, generally red to indicate an error condition or blue to indicate a value that has changed from the last update. However, this type of display is mostly useful in identifying a particular subsystem with an anomalous condition to aid in focussing on a particular problem. For example, for several weeks during the mission, the rover Spirit was having trouble with its right front wheel. The problem manifested itself in excessive current draw by the drive motor on that wheel and this was shown on the numeric displays immediately. However, it took examination of time histories of current draw over several weeks to track the progression of the problem, determine probable life expectancy of the motor, and suggest mitigation activities. One such activity was to drive the motor back and forth while warming it with its internal heater in an attempt to redistribute the lubricant. This had minimal effect so usage of that drive wheel was curtailed in order to extend its life. The rover was driven backwards while alternately dragging and driving the wheel, with a 10% duty cycle, to maximize the range available. This method was used successfully for several weeks until it was noticed that the current draw had dropped back to normal and regular usage of the wheel was authorized.

From the previous example, it can be seen that the use of time histories of state information is critical to analyzing many issues. Many types of plotting tools are available and RSVP includes a set for analyzing a variety of mobility issues. The channels that can be displayed in RSVP pertain almost entirely to mobility and arm operations and include XYZ position, orientation, overall tilt to investigate tilt limit issues, northerly tilt to examine incident solar energy issues, etc. Figure 3 shows a display of several mobility channels for analysis. RSVP makes no distinction between state information collected onboard the rover during operations and state information collected during simulation and rehearsal of planned operations. Thus, the data can be displayed side-by-side for rapid comparison of planned and actual activities of the rover.

State information from actual rover activities is typically used to analyze the behavior of the rover during traverses, especially when using its internal hazard avoidance system. The rover may arrive at its intended destination through a circuitous route that expends additional energy and puts additional wear on components when a simpler route might have been available. Understanding the rover's hazard avoidance software's interactions with various terrain types aids the operators in planning optimal traverses. A particular example was a planned traverse of about 40

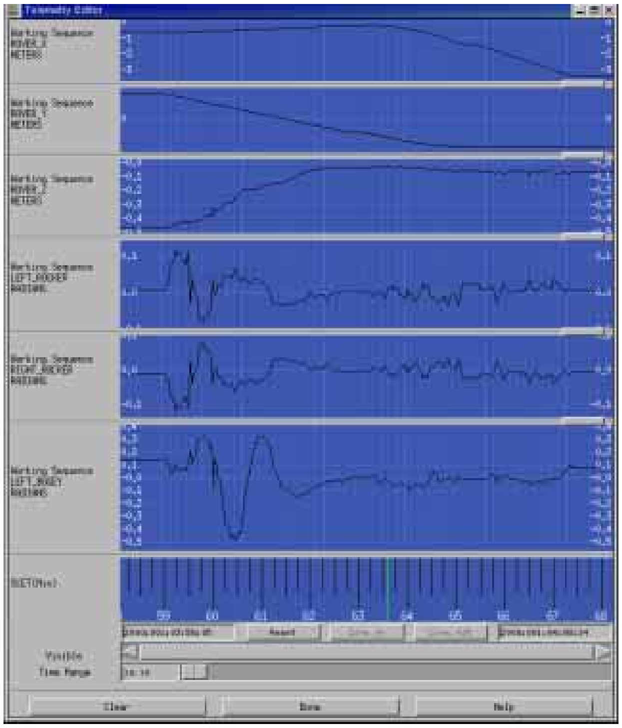


Figure 3 – Graphical Display of Rover State Information

meters across a rocky terrain. As the rover neared a cluster of rocks, it identified the rocks as hazardous and attempted to find a path to the left of the cluster. It drove forward and backward, slowly turning left, without finding a clear route to the left until it had turned far enough for the rear of the rover to be pointing to the right of the cluster. The rear cameras imaged the terrain, identified a clear route, and the rover proceeded backwards around the rock cluster and the rest of the way to the goal, only turning to face the destination at the last moment. Examination of final state would show that the rover reached the destination. Examination of the intermediate route would show the interesting behavior and suggest possible terrain types to avoid and preferential routing through problematic areas.

Most state information is subject to noise and error and expected levels must be kept in mind when analyzing the data. Some sensors may fail or occasionally give erroneous readings. When a sensor is recognized to be giving erroneous readings, its state information is removed from the state history to avoid contaminating the data and making analysis more difficult. However, small errors can creep in unnoticed. These are often detected when visualizing the rover in its current state in the context of the terrain model. Errors often show up as wheels not in contact with the surface, either hovering or submerged. This type of display will lead to further investigation into the state data to identify any erroneous data.

Other errors are simply ignored. For example, when the rover is traversing across the surface, the wheels may slip, the entire rover might slide downhill, or other actions might occur that throw off the internal dead reckoning of the rover. Even without slip, the dead reckoning is subject to some error. The typical response is simply to take new imagery at each stop and build new terrain models (see sections 2.2 and 2.3 for more details). Since the images and models are relative to the current rover position, they are perfectly acceptable for use for planning additional traverses.

In the case where excessive slip would be problematic, such as a location with a hazard downslope from the planned traverse track, visual odometry can be used to reduce the errors in dead reckoning due to slip. Visual odometry is a process where images are taken by the rover of the terrain prior to a move. After the move, usually less than 60cm or so, another image is taken of the same terrain region. The images are compared and features are tracked from one image to the next, and the actual motion is computed. This process is repeated for a succession of steps towards the local goal point. This method reduces the error in location to approximately 1%, or20cm in a 20m drive. However, the process may not converge on a solution if the terrain is feature-poor, the images do not have sufficient overlap, or other problems exist for the feature tracking. In this case, the dead reckoning solution is used, which may be inaccurate. In the case of serious hazards, drive sequences are set up to abort if the feature tracking fails to converge too often.

Arm activities of the rover rarely need to be analyzed this way because the flight software for controlling arm operations is integrated into RSVP, as discussed in more detail later. Thus, arm activities that successfully simulate in RSVP almost always work as intended in operations. The exceptions are due to inexact knowledge of the terrain, which leads to insufficient knowledge of actual arm-terrain interactions. To overcome this, a range of contact positions of instruments and terrain are simulated and commanding adjusted to avoid problematic areas.

In nominal operations, the primary state information that must be conveyed to RSVP is the final or current state of the rover. This includes mobility (XYZ position, orientation, joint angles, steering angles, etc.), IDD (joint angles, currently active instrument, contact switch states, etc.), error conditions (unable to reach goal, unexpected contact switch trip, etc.), and commanded preclusions (IDD precluded or mobility precluded). This information is distilled from the downlink data stream by the Mobility/IDD downlink analysts and provided to the rover operators. Additional information is also needed for planning in some circumstances. For example, drift occurs in the Inertial Measurement Unit (IMU) during operations that must be corrected by tracking the sun. The drift accumulates and the sun tracker must be activated about every 10000 seconds of activity time for the IMU. The rover operators provide estimates of IMU activity to the Tactical Downlink Leads who combine it with prior activities to determine when a sun tracking operation must be commanded. Since resetting the rover's orientation, which the sun find does, can invalidate planned rover activities, particularly IDD activities, the sun tracking operation is commanded by the rover operators in conjunction with their activities to minimize disruption. Thus, engineering requests for such operations are also provided to the rover operators as a part of the rover state.

2.2 Image Browsing

The MER rovers each carry a total of nine cameras, or imagers, as shown in Figure 4. Each camera has a CCD array of 1k x 1k pixels with 12 bits of resolution per pixel. They are arranged in four pairs of stereo imagers, with the ninth being the Microscopic Imager as one of the instruments on the IDD. Each type of imager has a primary use and the field of view and mounting take that into account.

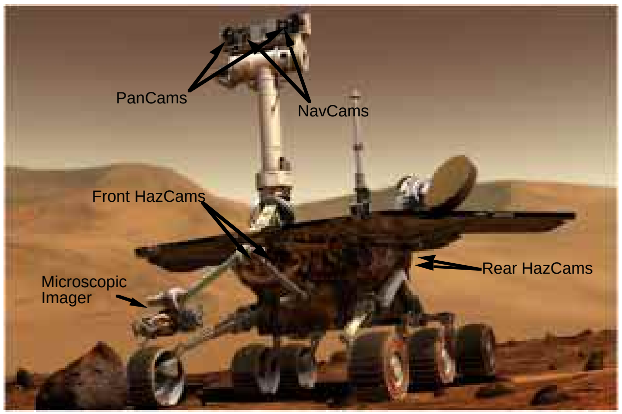


Figure 4 – Rover Cameras

The HazCams, or Hazard Avoidance Cameras, are mounted in fixed positions below the deck front and rear for examining the terrain within two to four meters of the rover's current location. They have a wide field of view and are tilted down such that the nadir is within the field of view. They are used both for analyzing the terrain for hazards and for imaging the terrain to be interacted with by the arm instruments. The fixed mounting means that there is no pointing error that must be allowed for when analyzing the imagery returned from these instruments.

The NavCams, or Navigation Cameras, are mounted on the mast above the deck and may be slewed and tilted to point in almost any direction. Each has a field of view of about 45° and is monochrome, like the HazCams. The NavCams are primarily used for tracking features during traverses for visual odometry and for imaging nearby terrain for three-dimensional modelling and traverse planning.

The PanCams, or Panorama Cameras, are also mounted on the mast above the deck. Each has a field of view of about 16° and contains a filter wheel for capturing multispectral imagery. The PanCams are primarily used for science observations in the multispectral domain but PanCam imagery is also used for modelling of farther terrain for long-range traverse planning.

Each stereo pair captured by any of the imagers is processed on the ground to utilize the stereo information to compute the shape of the features being viewed. The pair of images are correlated to find matching pixels and compute the disparity, or positional offsets, of each matched pair. The disparity information, along with the camera information, is used to compute the range to the feature displayed within the pixel. Finally, the rover position and orientation and the mast orientation are used to compute the (x,y,z) location of the feature. Since each feature is actually

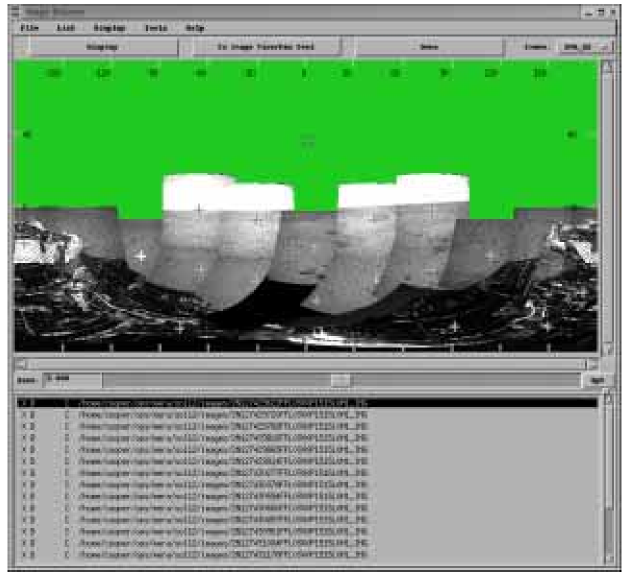


Figure 5 – Image Browser Display

associated with a pixel, a range image is generated with each pixel containing the range to the feature visible in that pixel. Similarly, an XYZ image is generated with each pixel containing the (x,y,z) location of the feature visible in that pixel. For more details on this process see [ALEX05], [DEEN04], and [DEEN05].

Generally, several PanCam, NavCam, and HazCam image pairs are captured each sol during operations. This imagery may be reviewed and analyzed using the Image Browser tool of RSVP. The Image Browser reads in the specified imagery and displays it in context with other imagery, using camera model data stored in the image files to properly mosaic the images. Figure 5 shows an example of the display of the Image Browser with several NavCam and PanCam images. Each image is displayed with an optional boundary and center position mark. In this example, an entire panorama of NavCam imagery has been displayed, along with several PanCam images in the expected traverse direction. The Image Browser can zoom and pan to closely examine any area of the display. The images can be displayed in Rover Frame, with azimuth and elevation relative to the rover, or in Local Level, with azimuth and elevation relative to local North and the horizon. This is useful for specifying targeting information for pointing the cameras at specific targets for additional imaging, as well as providing a common reference frame for indicating targets and drive direction. Individual or multiple images in the display may be selected for operations such as applying a color map to brighten the images or transmitting the images to the Stereo View tool.

When loading in images, the corresponding range and XYZ images are loaded as well. This allows the Image Browser to be used to analyze range, slope, roughness, and other information for planning traverses. It also provides the capability of specifying a target and computing the surface normal. Many RSVP activities utilize targets specified by the Image Browser, or imported from the Science Activity Planner, for building command sequences.

In addition to the Image Browser, RSVP includes an additional image viewing tool, the Stereo View tool. This tool accepts image information from the Image Browser and displays the corresponding left and right stereo images, either individually or in stereo mode for use with CrystalEyes® glasses for an immersive effect. Use of the stereo glasses, as seen in Figure 6, presents the operator with the least processed imagery with the highest fidelity for rapid understanding of the local terrain conditions. Small ridges that occlude hidden areas become quickly apparent in the stereo view and surface roughness and rock clusters are easy to understand. One twist is that the difference in the field of view of the PanCam, NavCam, and HazCam imagery presents challenges to the viewer to correlate what is seen with reality. Specifically, the narrow field of view of the PanCam gives the sense that faraway features are up close while the wide field of view of the HazCam does the opposite. Often the solution is to move the head closer to or farther from the screen such that the display window approximates the camera field of view relative to the eye.

2.3 Terrain Modelling

One drawback of the Image Browser and Stereo View tools is that their view is limited to that of the cameras on the rover. It is quite easy to tell if one rock is closer to the rover than another rock. It is much more difficult to see if the rover will fit between the rocks. Essentially, the operators need to be able to visualize the rover within the terrain and analyze interactions between the rover components and the terrain. Unfortunately, the rover cannot stand back and capture imagery of itself. To solve this problem, the range and XYZ information generated from the stereo imagery is used to build three-dimensional terrain models for visualization and simulation purposes as described in [WRIGHT04] and [WRIGHT05]. Then a virtual rover model can be used to examine terrain interactions.

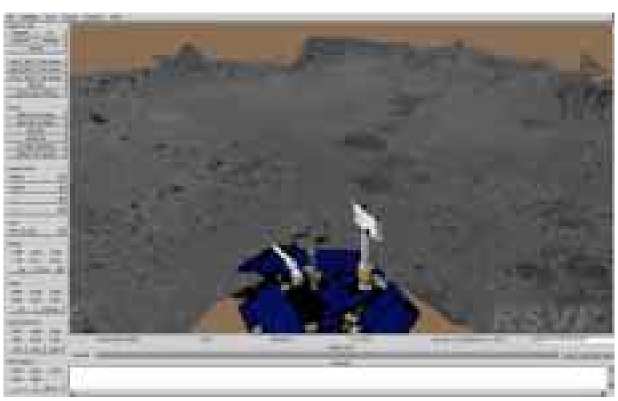


Figure 7 – Terrain Model Built from NavCam Imagery

The terrain models contain two primary components, a multi-resolution, multiple level of detail triangle mesh, and a height map. The mesh is used for visualization while the

Figure 6 – Use of Stereo Glasses

height map is used for computing interactions between the wheels and the terrain. Figure 7 shows an example of a mesh built using NavCam imagery taken near Spirit's landing site in Gusev crater as visualized in RSVP.

2.4 Terrain Visualization

Visualization of the terrain models is performed by the HyperDrive tool, which has several modes. The primary use is the flying camera in which the operator moves a virtual camera around the terrain model using a tabletop or free motion mode. Bringing the camera up to look down on the terrain from above allows the operator to quickly identify areas hidden from the rover as these will be unpopulated with terrain data and will appear as holes in the terrain mesh. It also allows the operator to quickly get a big picture view of the surrounding terrain. The center of the tabletop can be the rover or a target to aid in visualizing the terrain around them.

Several additional visualization aids can also be displayed within HyperDrive. These include a horizontal plane to easily locate the origin and identify up, cross, and down-slope directions. JPL's SPICE kernels are integrated with HyperDrive allowing visualization of ephemeris data showing the direction to various bodies in space including the sun, Earth, Mars' moons Phobos and Deimos, and various orbiting spacecraft. In addition, HyperDrive can also display shadows cast by the sun at any desired time, allowing the operator to visualize lighting and shadowing of targets during planned observations. Icons for planned commands can be overlaid on the terrain models as well as rover tracks from simulations or actual telemetry.

Of great importance is the ability to visualize interactions between a virtual rover model and the terrain. HyperDrive contains a rover model that becomes a cursor in one mode. The cursor is dragged over the terrain using the mouse, and rotated to any orientation, and the wheels, along with the rocker and bogey suspension, track the terrain model in realtime. Figure 8 shows an example of the middle wheel of the rover being articulated over a rock. The rover-terrain interactions are computed using the height map for easier sampling in the kinematic simulation.

Figure 8 – Articulation of Suspension by Terrain Model

Because HyperDrive does not distinguish between simulated and recorded rover state information, the same display can be used for real-time kinematic simulation and playback of either rehearsed planned operations or recorded state from actual operations. A VCR-like control allows the user to step through time and view the rover state recorded at each time step. In addition, the data can be played and the animation viewed in realtime or at any fraction of realtime.

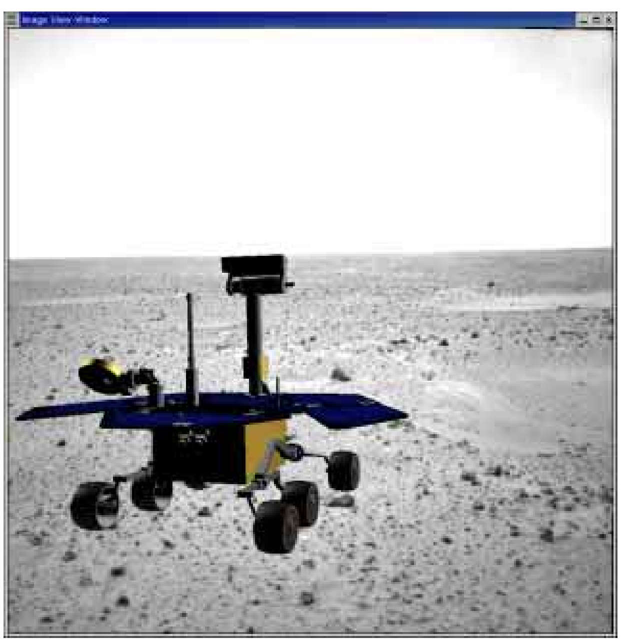
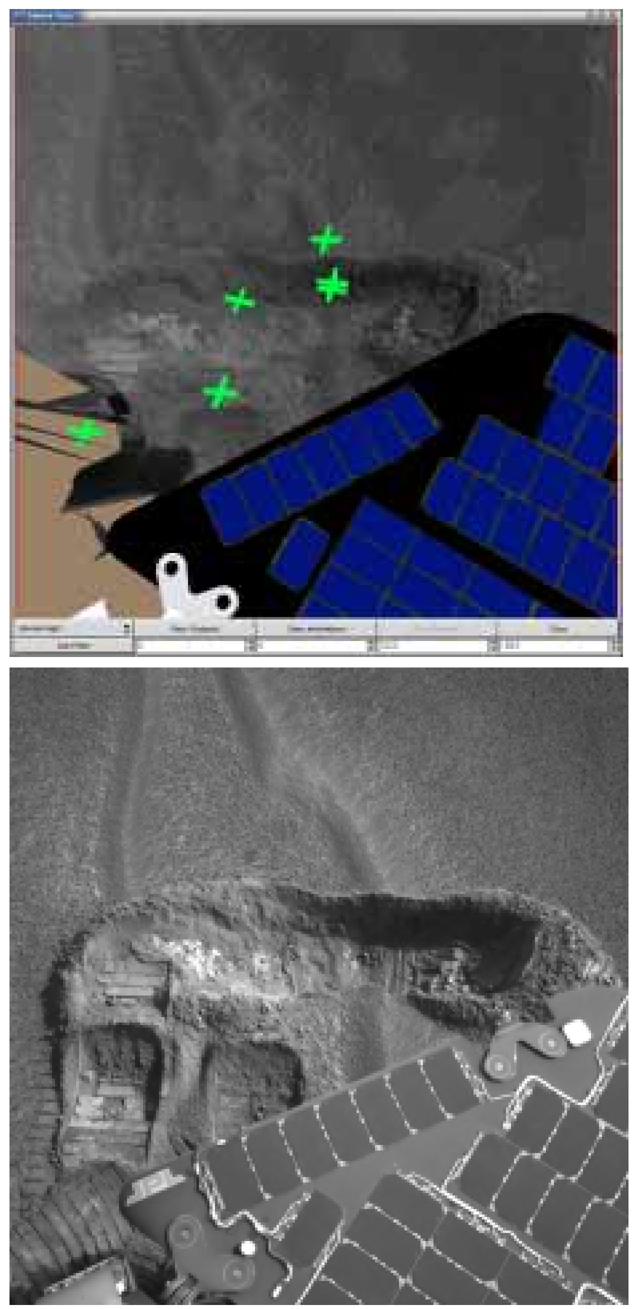


Figure 9 – Stereo View Window (Mono Mode) with Rover

The underlying framework for the three-dimensional terrain visualization is the OpenPerformer® library. This framework supports modelling and visualization of multiple objects and multiple views into the world containing those objects. This mode is used for the Stereo View tool described above and it allows the rover model, the command icons, the rover tracks, and other visualization aids to be displayed within the images seen in the tool. This is especially useful in stereo mode where the three-dimensional terrain is enhanced with three-dimensional representations of these aids, including the three-dimensional rover model. Figure 9 shows an example of the rover model overlaid on the imagery in the Stereo View tool.

Another tool that visualizes the same environment is the Camera View window. This window displays the view from any of the cameras on the rover, including the Microscopic Imager on the IDD, from the current rover position and state. When planning observations with the cameras, it is quite useful to visualize what the camera is expected to see prior to the imaging. Often, this visualization is extremely accurate as seen in Figure 10. However, the accuracy is highly dependent on the quality of the terrain model in the vicinity of the feature or area being imaged.



2.5 Immersion and Telepresence

Full immersion in the visualization, giving a sense of telepresence on the surface of Mars, is a desirable long-term goal. Being able to interact fully with the Martian surface while planning activities might make the process faster and more intuitive and increase the amount of science accomplished during a mission. The RSVP system has gained from several years of work in this area [WRIGHT98]. However, the systems needed to provide such a telepresence are expensive and not yet capable of providing the fidelity and sense of immersion that would ultimately be desired. In some sense, the tools described above attempt to provide a fractional sense of being present on Mars. The Stereo View tool, with the stereo glasses, really does give the impression that the user is seeing Mars as though they were standing on its surface. The PanCam imagery, with its high resolution and multispectral capability, make features highly visible to the operator. The three-dimensional visualization features of HyperDrive give some sense of being able to explore the Martian terrain and wander about while the simulated interactions between the rover models and the terrain models give realism to the viewing of rover activities. The RSVP system has shown that even the limited sense of presence given here greatly enhances the ability to create command sequences and operate the rovers on a daily cycle and, in fact, enables the mission. However, full immersion and telepresence is still a long-term goal.

3. Sequence Generation

Once the current state of the rover has been analyzed and the surrounding terrain reviewed for potential hazards and targets of interest, it is time to begin building the command sequences for the current sol's activities. The rovers are autonomous in that they conduct their activities while out of contact with ground controllers. However, the true level of autonomy varies with the types of commands being executed. Some commands are very low-level and generally execute immediately. These include commands for such things as turning on heaters, testing conditionals, and calling subsequences. More autonomous commands include driving forwards one meter, during which many error checks are being constantly performed for tilt limit violations, current limits being exceeded, etc. The command with the highest level of autonomy is the go to waypoint command with hazard avoidance. This command performs terrain analysis, to identify hazards, and safe path selection to attempt to reach the commanded destination safely. This command is essentially nondeterministic because terrain knowledge on the ground is always inferior to knowledge onboard the rover.

Commands are bundled into sequences that resemble subroutines in a program. Each sequence is typically dedicated to a single activity such as mobility, IDD, imaging a particular target, etc. The rover operators are responsible for the mobility and IDD sequences, dedicated Project Uplink Leads (PUL) for each instrument create the sequences for performing specific observations and controlling specific instruments, and the Sequence Integration Engineers (SIE) provides the master and submaster sequences for the sol. The master sequence for the sol is responsible for the overall timing of events, including shutdowns, restarts, activation of submasters, deactivation of other sequences, etc. The submaster sequences generally control the activities for a block of science observations, mobility, or IDD activities that perform consecutively.

Sequences can run sequentially or in parallel. In fact, the master sequence is always running and all other sequences run in parallel to the master. This allows the master to terminate other sequences if they take longer than expected to execute and other activities are required to begin. However, subsequences can also run in parallel with each other. This allows such things as heating the drive motors in preparation for mobility activities while performing science observations on targets prior to driving away. Another interesting use of this capability is running an obstacle check sequence in parallel with a driving sequence. Thus, the drive need not pause to explicitly check for obstacles being too closely approached, which can happen in high-slip regimes. The obstacle check is being constantly performed and if an obstacle comes too close, further driving is precluded. This stops the drive from completing although it does not terminate the drive sequence.

The sequence generation process begins with the Science Operations Working Group (SOWG) where the science team meets to discuss the desired activities for the sol. The rover operators attend this meeting to advise the science team on constraints and feasibility of various scenarios and to gain understanding of the main goals and acceptable alternatives. The science team members, both local and located at home institutions, use the Science Activity Planner (SAP) to build the science plan for the sol and to mark targets of interest in the local terrain. Debate then ensues until the sol's plan is finalized, priorities are set, and targets selected. At this point, the meeting adjourns and work begins on the actual command sequences.

The next step in the process is the Activity Plan Approval Meeting (APAM). Prior to this, the Tactical Activity Planner (TAP) builds a detailed activity plan, with a timeline, for performing detailed resource and constraint analyses of data accumulation, power consumption, etc. Early in the mission, the scientists would enthusiastically oversubscribe the rovers and some activities would have to be removed from the plan to meet the constraints. More recently, additional tools have supported the scientists with earlier constraint information and few, if any, activities are usually cut from the plan. In addition to the activity plan prepared by the TAP, the rover operators prepare a skeleton of the command sequence for the sol's activities and generate an animation showing the physical activities of the IDD and/or mobility. During the APAM, the plan is reviewed and the animation displayed to get final approval of the plan and verify that the targets being utilized match everyone's understanding.

The third step in the process is the Sequence Walkthrough. At this meeting, the master and submaster command sequences and the rover mobility and IDD sequences are reviewed, command by command, by the science and operations teams. This meeting verifies that no activities have been missed and that the appropriate subsequences are being called to perform various observations at the correct times. Small changes are often made to the sequences during this process but generally they have limited scope unless some major element has been left out or later analysis proves that some activity is just not feasible.

Once the final versions of all the sequences have been delivered by the various teams, they are integrated by the SIE. The sequences are bundled and formatted appropriately for uplink, cross-checked for confirmation of delivery of all sequences used during the sol, and a variety of documentation and review products are generated. Then the Command Approval Meeting (CAM) is held, at which the science and operations teams again go through many of the sequences command by command to verify that the correct versions of sequences were delivered and integrated, that the sequences do what is expected, and that final modelling shows that all the various constraints are being met. The bundled sequences are then delivered for uplink.

3.1 Targeting

Targeting, or specifying a target of interest for science observations, IDD activities, or even traverse waypoints and obstacles, takes several forms. The primary forms of a target are location and pointing. The first, location, uses an XYZ location, coupled with a surface normal, to define a target. The location is specified in site frame, which is defined by a local origin generally within a few meters of the rover and the +X axis to the north, the +Y axis to the east, and the +Z axis down. The second form, pointing, uses an azimuth and elevation for pointing the mast cameras, the PanCam, NavCam, and the Miniature Thermal Emissions Spectrometer (mini-TES). Pointing can thus point at a location for ground observations or in a direction for sky and celestial object observations.

Location is the primary form for targets to be observed on the ground, particularly for mobility destinations and IDD targets. Images containing features of interest can be queried for the XYZ coordinates of pixels and these used for capturing new images. SAP uses this form and many targets are specified by the science team using SAP and imagery taken from previous sols. Additional targets are specified using the Image Browser as described above. There are a few concerns about using location targeting that the user must be aware of. One is that IDD targets are not specified in site frame but rather in rover frame. This coordinate frame has the origin at the center of the rover, and the +X axis to the front, the +Y axis to the right, and the +Z axis down. RSVP converts the coordinate frames for the user when generating IDD commands so this is generally not a problem but it does mean that the IDD commands typically cannot be typed in directly or the conversions will not be applied. In addition, the IDD targets typically require a surface normal. This is necessary because the instruments on the IDD have an area of contact that must be maximized to obtain the best science results. Thus, the surface normals need to be included as part of the target, and both SAP and the Image Browser do this. Another issue to keep in mind is the declaration of a new site. Typically a new site is declared, by issuing an increment site index command, when the rover's knowledge of its location within the current site has deteriorated. This occurs from driving in high slip regimes without visual odometry or when the rover has traversed near the edge of or out of the terrain information of the current site. However, once a new site is declared, the rover is now located at the origin of the new site and any commands using coordinates in the previous site are now invalid. Thus, it is necessary to coordinate the incrementing of the site index with the instrument PULs. Since imagery taken after a traverse needs to be in the new site frame for targeting future observations, the procedure is generally to perform the traverse, perform observations specified in the previous site frame, increment the site index, then perform observations in the new site frame or using pointing targets.

Pointing targets are used for many types of observations. For example, atmospheric and meteorological observations are specified using pointing targets. In addition, pointing at the sun to perform tracking for attitude refinement or to observe transits by Mars' moons is done in pointing mode with azimuth and elevation. Most importantly, mosaics are specified by pointing also. Mosaics can be small, such as a 2x1 mosaic of an interesting target, or large, such as a complete NavCam panorama of two rows of ten images each. The NavCam panoramas are typically captured in order to build terrain models for additional, upcoming traverse planning. Subsets of the complete panorama may be specified and captured when downlink bandwidth precludes receiving all the data in time to produce the models and plan the next traverse. In such a case, the expected drive direction is estimated and used as the center azimuth of the mosaic.

Both location and pointing targets are used for mobility activities. Location targets are often used to specify waypoints, destinations, and obstacles. Commands exist for computing the distance from the rover to a specified location and then performing conditional commanding based on the distance. These include precluding driving if too close to an obstacle, as mentioned above, or bumping towards the destination if still too far from it. Turn commands can be specified to turn to face towards or away from a specified location.

Pointing targets are used less frequently, most often with a turn to an absolute heading to enable improved communications performance or to alter shadowing of the rover and its equipment by the camera mast.

3.2 Rover Sequence Editor

The Rover Sequence Editor (RoSE) is the backbone of the command sequence generation process. RoSE contains the master view of the sequence, maintains the unique identifiers for identifying selected commands in all the various tools, and distributes command and simulation updates throughout RSVP. Figure 11 illustrates the appearance of the RoSE interface. At its most fundamental, RoSE is a text editor for entering commands in sequence and for editing existing command sequences. However, RoSE provides many features that aid the operator in rapidly producing valid sequences. It is used by every operator, SIE, and PUL to produce the command sequences for delivery, documentation and archive products for review, and the final command bundles for uplink. Details outside the scope of this paper can be found at [MAXWELL04] and [MAXWELL05] but some of the highlights will be touched on here.

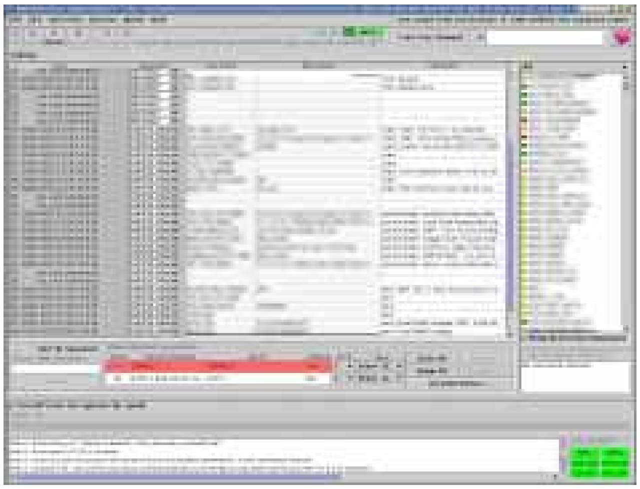


Figure 11 – RoSE Interface

3.2.1 Macros

When building the actual command sequences, RSVP offers significant assistance for the rapid production of validated sequences. On Mars Pathfinder, the total size of the sequences sent to the rover were around 100 commands [COOPER98]. On MER, the sequences can total over 1000 commands for a single sol and mobility and IDD sequences have exceeded 500 commands on occasion. Building and validating such large sequences requires support from the tools. Many sequences have similar blocks for performing functions that must be done each sol or for each activity. For example, prior to using the IDD, the temperature sensors on the arm joints must be tested and temperatures set to provide for proper current limit measurements. Since this must be done prior to all IDD activities and is done the same way each time, the commands for the process are stored in a macro and expanded when desired. Similarly, when ending the IDD activities, the recorded activity information, the state history for the activity, must be saved and then compressed for downlink in the same way every time. A macro encapsulates this behavior as well.

The macros provide a simple way to specify a few parameters and generate many lines of valid commands. In addition, the macros make it more difficult to miss a critical command in a large block of commands if the block was generated entirely from a single macro. The macros also provide templates for activities in which many commands are present and markers indicate a required additional command that must be generated separately. As mentioned above, IDD targets are specified in site frame while the IDD commands to access those targets are issued in rover frame. Rather than the user converting the coordinates, the HyperDrive tool is used to specify the desired site frame target, convert the coordinates, and generate the IDD commands. These commands are then inserted into the command block generated by the macro at the designated locations.

3.2.2 Sequence Completion

As mentioned above, macro expansion produces large portions of a sequence for most IDD and mobility activities. However, the specific motions that depend on target locations and directions, or on hazard avoidance, must be generated with knowledge of the terrain and the targeting information. These commands are generated using the HyperDrive tool. All the tools in the RSVP suite are tightly coupled with a shared view of the command sequence being developed. As previously mentioned, RoSE maintains the master view of the sequence and communicates this view to all the other tools when any change is made. Changes can be made in RoSE by editing a command, expanding a macro, or entering a command from scratch. Changes can also be made in HyperDrive, generally inserting commands dependent on the three-dimensional information conveyed in the visualization tools. Command insertion requests from HyperDrive are communicated to RoSE, which performs the insertion and broadcasts the inserted commands to all the tools. While HyperDrive contains the entire command sequence under development, only the commands with a visual component are visualized. This would include waypoints, turns, and IDD movements but not heater commands, switch state testing, or other commands not coupled to the terrain. Figure 12 shows how HyperDrive uses a variety of icons to represent the commands and their relationship to the terrain (Figures 7 and 8 show the overall HyperDrive user interface).

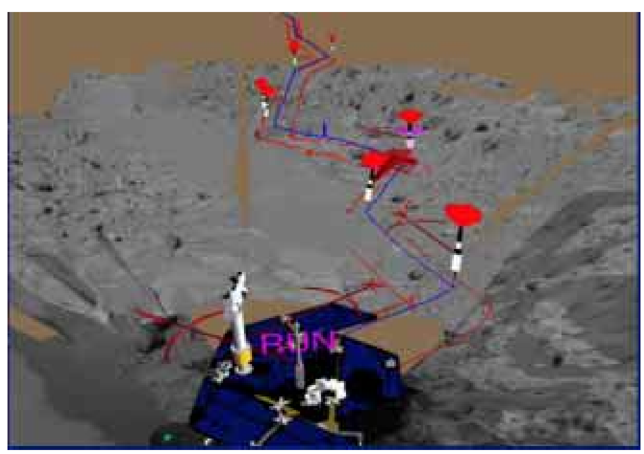


Figure 12 – HyperDrive Terrain and Icon Display

3.3 Sequence Validation

Sequence validation is a multi-tiered process. Validation within RSVP ranges from syntax checking of commands and range-checking of arguments through simulation of the commands via software and on up to running the commands through the flight software itself or onboard an actual rover in the testbed. These steps verify that the sequence is syntactically correct and would execute on the rover. Flight rule checks verify that a variety of rules are not being broken. These might verify such things as that the dust cover on the Microscopic Imager is opened prior to taking an image. This provides additional feedback on the validity of the sequence. An additional level of check is the rehearsal and playback of the sequence for visual verification that the sequence performs correctly and interactions with the terrain are as expected. Finally, if some major component of the sequence has never been done before, it is often run on an actual rover in the JPL testbed to verify proper behavior prior to utilizing it on Mars.

Syntactic checking of the commands is performed continuously. RoSE ingests the command dictionary at startup time and builds the user interface for entering commands and checking parameter types and ranges automatically. Invalid commands or parameters are immediately highlighted to alert the user.

3.3.1 Sequence Simulation

Sequences are simulated almost continuously during development. As the user is adding commands to the sequence, or editing or deleting existing commands, the sequence is sent to a tool called SEQGEN, developed at JPL and adapted to each mission and spacecraft. SEQGEN performs simulation of the sequence and returns the time at which each command is expected to execute, if ever, including reporting of how many times each subsequence was called and which commands executed during each subsequence execution. The durations of most commands are known a priori and these durations are used for the simulation. However, SEQGEN does not have information on the terrain or the rover state so it is unable to correctly estimate the duration of mobility and IDD motion commands. This part of the simulation process is performed by HyperDrive.

3.3.2 Sequence Rehearsal and Playback

HyperDrive performs simulation of sequence execution through links to actual flight software for IDD activities and through kinematic simulation tools for mobility activities [YEN05]. HyperDrive can work in conjunction with SEQGEN or standalone. When working with SEQGEN, the report generated by SEQGEN is used as an estimate of when each command will execute, even if multiple times. It is an estimate because the duration of the IDD and mobility commands is unknown. HyperDrive, which is aware of rover state, then performs a simulation of the IDD and mobility commands and returns the expected durations to SEQGEN. This initiates another simulation by SEQGEN, which produces new estimates of start time, and this stimulates another simulation in HyperDrive. After a few iterations, the simulation has converged on a specific set of start times and durations for all commands expected to execute. HyperDrive can perform a simulation without SEQGEN by assuming a particular path through the sequence but the commands that do not interact with the terrain are modelled very simply. Thus, it is preferable to utilize SEQGEN in the loop.

Once the simulation has completed, it can be played back to visually verify the proper behavior of the sequence. A VCR-like control widget is brought up and used to control the playback of the sequence as an animation in the HyperDrive window. As the rover model performs the commanded activities, the operator can view the animated model from any angle to verify that the behavior is as expected. The operator can stop and start, single step forward and backward, rewind, and fast forward to appropriately visualize the rover behavior. Note that this interface is identical whether viewing the results of a simulation or recorded state information from the actual rover. Thus, it is quite easy to visually compare the behavior of the simulated rover and the actual rover.

3.3.3 Flight Rule Checking

As the sequences being developed are written to files, the flight rule checker can be run to analyze the sequences and verify that a variety of rules are not being violated. For example, one of the Mossbauer Spectrometer switches on Spirit sticks occasionally. Often, when placing the Alpha Proton X-ray Spectrometer (APXS), the Mossbauer is used to touch the soil, then a tool change command places the APXS where the Mossbauer previously touched. The specific action of this command is to retract the Mossbauer, a move which expects the switch to start in the contact state and end up in the non-contact state, then a rotation of the turret to point the APXS in the right direction, a move which expects the switch to be in the non-contact state at all times, then a placement of the APXS, a move which expects the APXS contact switch to trigger. If the Mossbauer switch sticks during the retraction, and the turret rotation tries to start while it is still stuck, the move faults out and all succeeding IDD activity is precluded. To avoid this problem, the Mossbauer switch is disabled for all activities except the placement of the Mossbauer in contact with a rock or soil target. Essentially, the switch must be enabled just prior to placement and disabled immediately after placement. However, the movements to place the Mossbauer must be commanded through HyperDrive rather than through a macro so it is easy to forget the necessary switch settings. The flight rule checker verifies that, for Spirit only, any Mossbauer placement moves are bracketed properly by the switch commands.

Other flight rules include one to verify that all conditional commands include a comment stating "[assume true]" or "[assume false]" to identify the nominal path through the sequence, and one to verify that the appropriate initial state information has been loaded.

The flight rule checker runs as a standalone application but a future version will integrate it into the RSVP tool suite for continuous monitoring of the sequences as they are being developed. Use of the flight rule checker has shown the value of such a tool, with an adaptable set of tests that can be modified during the mission as new constraints become apparent and earlier constraints are shown to be unnecessary. The current flight rule checker is written in Perl and is relatively easy to extend. However, it may be preferable to have a flight rule specification language that more clearly represents intent. This is work for the future.

4. Archival Product Generation

A variety of documentation and archival products must be generated for each sol of activities. Some are used by the remote teams for review purposes, some are used by the downlink teams to know what to expect from the sol's activities, and some are archived for historical purposes.

4.1 Sequence Report Generation

RoSE prepares sequence reports at various stages of the planning process. A sequence report is an HTML document that lists all the commands in the sequence or sequences under development, includes comments and notations, lists the simulated command start time with repeated time entries if the command is executed multiple times, and error reports and flight rule checks performed by SEQGEN. Hyperlinks are included for rapid perusal of the desired information in the report. These reports are prepared for the master/submaster walkthrough and for the command approval meeting and aid the remote teams in following the discussion and presentation of the sequences. They are also archived for review to verify that the commands that executed were the planned ones in case of any anomalous behavior.

4.2 Animation Capture

HyperDrive supports the capture and export of animations in the form of MPEG files as well as state history files. Once HyperDrive has performed a simulation, the planned rover activities have been captured in the form of a state history. This state history may be written to a file as Rover Kinematic State Markup Language (RKSML), an HTML derivative. It may be read back in to drive a real-time animation in HyperDrive and is of the same form as the downlinked state history information for the rover's actual activities. HyperDrive can ingest several state history files and simultaneously animate a rover model based on each file. This allows the comparison of planned and actual activities in their behavior and timing. Once the sequence has been finalized and approved, the rover operators generate and store an RKSML file for the sol in the report archive. This file is also used for a variety of other purposes. The downlink team often reviews the planned activities by viewing an animation from the RKSML file to familiarize themselves with the expected telemetry. The RKSML file is also analyzed by a specialized tool to verify that the turret of the IDD is not rotated too far, which can inadvertently close the doors of the APXS, and to compute the closest expected approach to all obstacles specified in the sequence.

In addition to the RKSML, HyperDrive can also capture a sequence of frames during an animation of the planned or actual activities. These frames are written to disk as the animation is rendered and later converted to an MPEG file to be attached to the uplink report. This allows the viewing of the planned activities via a web browser rather than having to use RSVP, although the flexibility of camera placement is lost.

4.3 Uplink Report Generation

The uplink report is an HTML-based report that contains a description of the planned activities for the sol. A standalone uplink report generator creates the report by extracting marked comments from the sequence files, state information from the RKSML file, and file information from the repository. The uplink report contains an overall summary of the sol's activities, a pseudocode summary of the behavior of each sequence, the initial rover state at the start of the sol, the rover state at the end of each sequence, the final rover state, a history of instruments used on the IDD, joint motion ranges for the IDD, and links to auxiliary information such as the RKSML and MPEG files.

The uplink report is used both by the downlink team and the uplink team for the next sol. The downlink team wants to know the expected state of the rover at various times during the sol to compare to the actual telemetry, as well as a general understanding of the planned activities. The uplink team reviews the previous team's reports to familiarize themselves with the history of activities at the current location.

5. System Architecture

RSVP's high-level design is reminiscent of a microkernel-based operating system: it has a central message-passing hub, and component applications plug into the hub and use it to communicate. Figure 1 illustrates the message-passing layer in RSVP and how it maintains a common view of the sequence. When, for example, you add a command in one RSVP component application, that application sends a message containing a chunk of XML that describes the new command. When other applications receive this message, they update their internal state (and their display) to include the new command.

RSVP's hub-based design yields several benefits. First, it simplifies multi-language development, allowing us to use C++ where performance is critical and Java or other languages elsewhere. In addition, the hub-based design facilitates the development of the overall system. Component applications can be developed and tested standalone; each application communicates to the rest of the system only via the hub, which is easily stubbed out during development. (This point was borne out in practice when we first connected RSVP's major components -- they worked together immediately, with only one small, easily fixed bug in one application's interpretation of a single message type. The whole thing was working flawlessly in a single afternoon.)

Third, the hub-based approach simplifies adding new functionality to the system. For example, integrating the Image Browser tool into RSVP only required adding one line to each of two configuration files.

Finally, because messages may be sent over the network as easily as within a single workstation, RSVP component applications may easily be spread across multiple workstations for crude load-balancing or collaboration. This capability has not been needed in operations but has been successfully demonstrated.

6. Conclusion

The RSVP suite of tools has been a critical factor in the success of the MER missions. It would be impossible to prepare the large command loads each sol without the capabilities that it possesses. It has proven to be a robust, easy to use, and capable of answering key questions about sequence validity and constraints. Certainly training is required to use RSVP but this is primarily in the general area of command sequencing and rover operations. Once these concepts are understood, RSVP feels natural for building sequences.

RSVP has met its prime requirement of supporting rapid assimilation and understanding of the terrain and operational constraints, rapid sequence generation and validation, and production of documentation and archival products. This can be seen in the very limited number of sols lost due to errors in the command sequences. The success of the MER mission and the tremendous amount of science data collected attest to the capability of RSVP.

7. Future Directions

There are several levels of future work to be done on the RSVP tool suite and the processes that utilize them. The lowest level is the addition of new features while keeping the basic paradigm the same. This level includes integrating the flight rule checker into RSVP rather than being a standalone program. A similar feature would be incorporating the uplink report generator into the suite. Because of RSVP's distributed architecture, it is straightforward to integrate new tools, as long as they adhere to the message-passing protocols used by the central core. It is fortunate that the developers of RSVP are also the prime users as this will assist in bringing lessons learned in operations back into the software for better usability.

Another level of future work is the extension of the tools into more immersive environments. RSVP provides only a partial sense of immersion when using the stereo glasses and when viewing the terrain models from selected angles. Additional hardware and software could provide a more immersive experience in support of telepresence. However, it is not known whether or not this will enhance the process of commanding and some research will be necessary in this area.

A third area of interest is in collaboration for building command sequences. The central core of RSVP supports multiple, distributed copies of HyperDrive, the Image Browser, and all other tools, except RoSE. These distributed copies can be run on distant machines and the command sequence manipulated in all locations. While this may lead to race conditions and other typical consequences of parallel operations, it also may lead to enhanced commanding, with reduced times to complete sequences and more rapid production of valid sequences. The current mission has utilized parallel development only by splitting the sequences logically and having multiple operators work on separate sections. More research is needed to see if a collaborative development effort would have benefits.

Another area of future work is the support of upcoming missions. One such mission is the Mars Science Laboratory. It is a rover with six wheels and multiple robotic arms, which makes it very similar to the MER rovers. However, a new rover is under development which is six-sided, has six legs, each terminated with a wheel, and the ability to roll over smooth terrain and walk over rougher terrain. Because the rover has six identical sides, it has no true forward direction. This means that the type of commanding for this rover will be very different from MER. It remains to be seen whether or not this rover can be supported in the existing RSVP framework.

A longer term area of future work is moving up the ladder of onboard autonomy. The MER rovers have commands to perform such autonomous functions as “Turn To A Rock,” where the rover selects a nearby rock and turns to face it for possible IDD operations. A “Go and Touch” capability is also under development to allow the rover to autonomously approach a rock to within IDD range, deploy the IDD, and place an instrument on the rock, without human intervention. These are relatively simple extensions of the MER rovers' capabilities but future systems will have higher levels of autonomy, with goals to achieve, autonomous onboard evaluation of the science value of local targets, interaction of instruments, and other features. These will require a more advanced type of commanding with goal specification and constraint evaluation. Research is needed to see how this will play in the RSVP system.

Acknowledgements

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[ALEX05] Alexander, D., P. Zamani, R. Deen, P. Andres, and H. Mortensen, “Automated Generation of Image Products for Mars Exploration Rover Mission Tactical Operations”, 2005 IEEE International Conf. on Systems, Man, and Cybernetics, Waikoloa, Hawaii, October 2005.

[BIESIADECKI05] J. Biesiadecki *et al.*, "Mars Exploration Rover Surface Operations: Driving Opportunity at Meridiani Planum," Proc. IEEE-SMC, Waikoloa, Hawaii, October 2005.

[COOPER98] B. Cooper, "Driving on the Surface of Mars Using the Rover Control Workstation," Proc. SpaceOps '98, Tokyo, Japan, 1998.

[DEEN04] R.G. Deen, D.A. Alexander, J.N. Maki, "Mars Image Products: Science Goes Operational", Proceedings of the 8th International Conference on Space Operations (SpaceOps), Montreal, Canada, 2004.

[DEEN05] Deen, R.G. and J.J. Lorre, “Seeing in Three Dimensions: Correlation and Triangulation of Mars Exploration Rover Imagery”, 2005 IEEE International Conf. on Systems, Man, and Cybernetics, Waikoloa, Hawaii, October 2005.

[HARTMAN05] F. Hartman *et al.*, "Data Visualization for Effective Rover Sequencing," Proc. IEEE-SMC, Waikoloa, Hawaii, October 2005.

[LEGER05] C. Leger *et al.*, "Mars Exploration Rover Surface Operations: Driving Spirit at Gusev Crater," Proc. IEEE-SMC, Waikoloa, Hawaii, October 2005.

[MAXWELL04] S. Maxwell *et al* “The Design and Architecture of the Mars Exploration Rovers Project's Rover Sequencing and Visualization Program," Proc. SpaceOps ‘04, Montreal, Canada, 2004.

[MAXWELL05] S. Maxwell *et al* “The Best of Both Worlds: Integrating Textual and Visual Command Interfaces for Mars Rover Operations," Proc. IEEE-SMC, Waikoloa, Hawaii, October 2005.

[WRIGHT98] J. Wright *et al,* “Immersive Environments for Mission Operations: Beyond Mars Pathfinder”, Proc. SpaceOps '98, Tokyo, Japan, 1998.

[WRIGHT04] J. Wright *et al,* “Terrain Modelling for Immersive Visualization for the Mars Exploration Rovers”, Proc. SpaceOps ‘04, Montreal, Canada, 2004.

[WRIGHT05] J. Wright *et al,* “Terrain Modelling for In-situ Activity Planning and Rehearsal for the Mars Exploration Rovers," Proc. IEEE-SMC, Waikoloa, Hawaii, October 2005.

[YEN05] J. Yen *et al.*, "Sequence Rehearsal and Validation on Surface Operations of the Mars Exploration Rovers," Proc. IEEE-SMC, Waikoloa, Hawaii, October 2005.